

Microsoft
.net developer
training

Create XML Web Services and Applications with Visual Studio .NET



The .NET Languages

Module 04 Agenda

- **Languages**
- **Visual Basic® .NET**
 - **Feature enhancements**
 - **Additional changes**
- **Visual C#™**
 - **Shape and structure**
 - **Statements**
 - **Built-in collections**
- **Visual C++® .NET**
 - **Managed extensions to Visual C++**

.NET and Languages

- **The .NET Platform is multi-language**
 - **Benefits:**
 - **Make code modules reusable**
 - **Make API access the same for all languages**
 - **Let developers use the right language for the task and share the code**
 - **All .NET languages are first-class players**
 - **Performance roughly equal**
 - **Can use all .NET features**
- **Microsoft is providing:**
 - **Visual Basic, Visual C++, Visual C#**

.NET Languages

- **COBOL**
- **Jscript .NET**
- **Eiffel**
- **Haskell / Mondrian**
- **Pascal**
- **ML**
- **APL**
- **Alice**

- **C**
- **Visual C++**
- **Visual Basic**
- **Visual C#**
- **Smalltalk**
- **Oberon**
- **Scheme**
- **Mercury**
- **Python**

Choosing a Language

- **Languages are true equals**
- **Language selection**
 - **Visual Basic or VBScript developer → Visual Basic .NET**
 - **Visual C++ or Java developer → Visual C#**
 - **JScript developer → JScript or Visual C#**

Languages

Demo

- **COBOL, Visual C# and Visual Basic .NET demo**
 - **Debugging**
 - **ASP .NET**

Visual Basic .NET

- **.NET integration**
 - VBRUN replaced by the .NET Framework
 - .NET Framework is available directly
- **Language innovations**
 - Classes
 - Inheritance
 - Constructors
 - Structured exceptions
 - Optional stricter type checking

Visual Basic .NET

Feature Enhancements

- **Structured exception handling**
- **Function overloading**
- **Free threading**
- **Namespaces**
- **Stricter type checking**
- **Parameterized constructors**
- **Class/Module statements**

Visual Basic .NET

Additional .NET Changes

- **Garbage collection**
- **Arrays are Object-based**
- **Variant DataType replaced by Object**

Visual Basic .NET

Demo

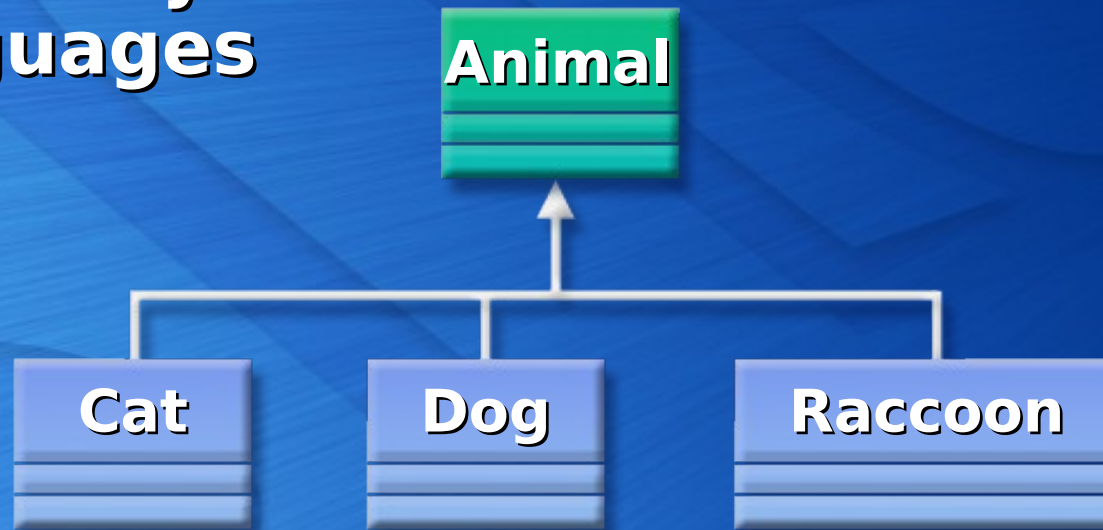
- Simple VB Application



Visual Basic .NET

Inheritance

- Visual Basic is a full OOP player
- Overloading - Polymorphism
- Same object features as other Object-Oriented Languages



Visual Basic .NET

Structured Error Handling

- **Try construct**
 - Guards code between Try and first Catch statement
 - Nested Trys to resume behavior
- **Catch statement(s)**
 - Filters errors
- **Finally block**
 - Always executes
- **Visual Basic .NET still supports “On Error GoTo”**

Visual Basic .NET

On Error Versus Try/Catch

fReRaise = False

On Error GoTo **ErrHandler**

<Code that may fail>

GoTo **Cleanup**

ErrHandler:

if **ConditionWeCanHandle** Then

<error handling code>

<error handling code>

fReraise = True

fReraise = True

End if

<clean up>

if **fReraise** Then **err.Raise err**

if **fReraise** Then **err.Raise err**

Try

<Code that may fail>

Catch *ConditionWeCanHandle*

<error handling code>

Finally

<clean up>

End Try

End Try

Visual Basic .NET

Demo

- **Error Handling**

Visual Basic .NET

Simplifications

- **Eliminates default properties without parameters**
- **Only one form of assignment**
- **Always use parentheses around arguments**

DoSomething(a,b)

- **Fix variable declaration syntax:**
Dim x, y as Integer
- **New operator same for all classes**

Visual Basic .NET

Reduce Programming Errors

- **Parameterized constructors**
- **Variables can be initialized where declared**
- **Circular references are no longer a problem**
 - **Garbage collector recovers instances in reference cycles**

Visual Basic .NET

Type Checking

- **Option for stricter type checking**
 - **Compiler catches more errors during compilation**
 - **Prevents implicit conversions that may fail at run time**
- **Explicit conversion (casting) operator: CType**
- **Enables support for overloaded methods**
- **Array declarations must specify rank**

Visual C#

Shape and Structure

- **First component-oriented language in the C/C++ family**
- **No header files, IDL, etc...**
- **Visual C# employs "definition at declaration" model**
 - Much like Visual Basic, Pascal, Modula and Java
- **Similar to Visual C++ "in-line" implementation**
 - Without implication on code generation
- **All code and declaration in one place**
 - Keeps code consistent and maintainable
 - Much better to understand for team collaboration
 - Declaration accessible through metadata
- **Conditional compilation but no macros**

Visual C#

Statements

- **Very C/C++: Flow control and loops**
 - **if (<bool expr>) { ... } else { ... }**
 - **switch(<var>) { case <const>: ...; }**
 - **while (<bool expr>) { ... };**
 - **for (<init>;<bool test>;<modify>) { ... }**
 - **do { ... } while (<bool expr>);**
 - **Structured error handling (Try/Catch)**

Visual C#

Collections

- **Straight-forward support for iterating over collections**
 - Can be used for arrays and other collections
- **Can also be used with any custom class**
 - Implements `IEnumerable` with `GetEnumerator()`
 - Returning object implementing `IEnumerator`

Visual C#

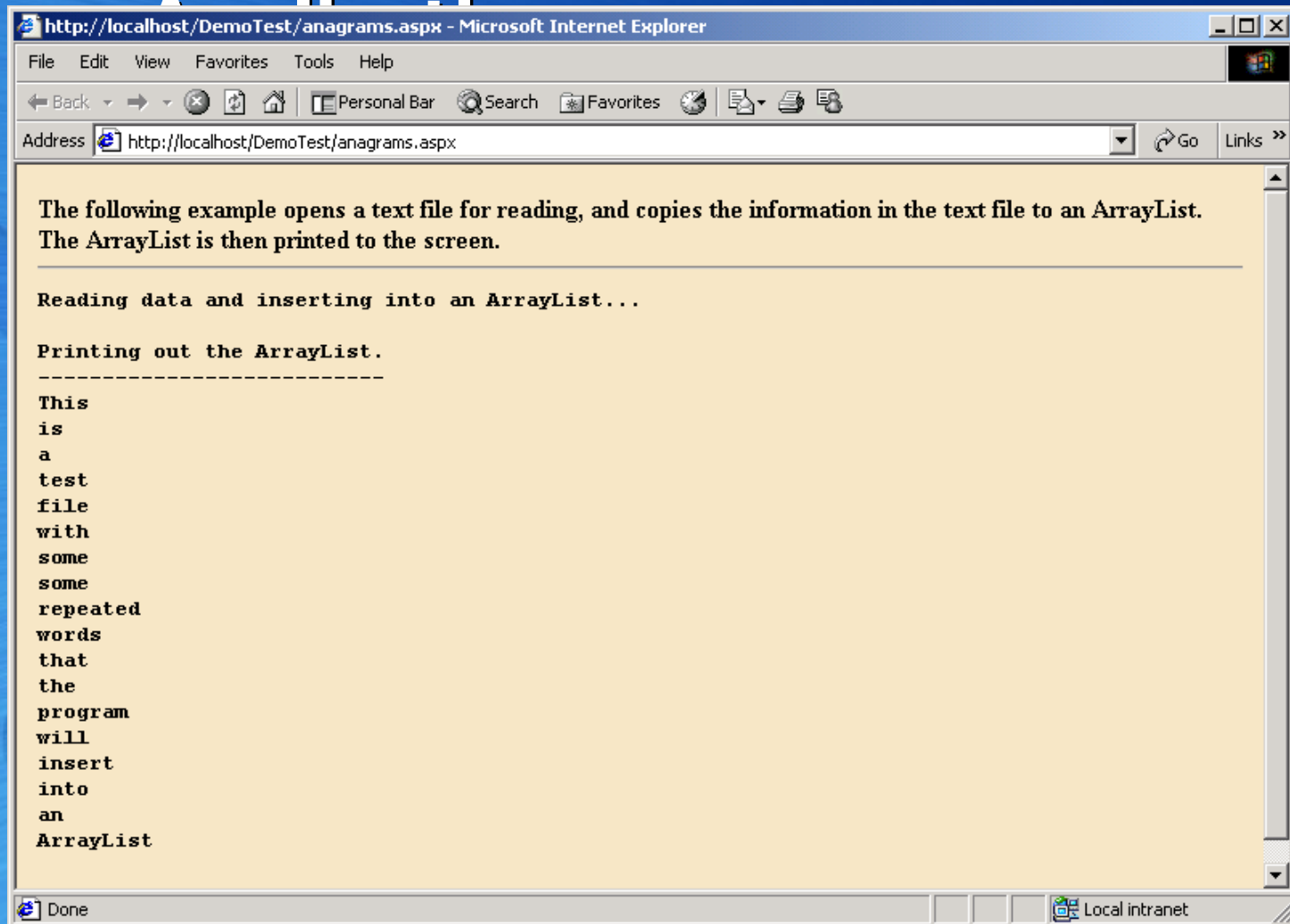
Operators

- **Very C:**
 - Logical/conditional: && || ^
 - Arithmetic: * / + - % << >>
 - Relational: == != < > >= <=
- **Not exactly C:**
 - For bool: & and | are logical with full evaluation
 - For integer: & and | perform binary AND/OR
- **Very un-C:**
 - is - Tests run-time type
 - as - Converts a value to a specified type
 - typeof - Retrieves run-time type

Visual C#

Demo

- Simple C#



Visual C++ .NET

- **Part of Visual Studio .NET IDE**
- **Visual C++ optionally targets .NET**
 - **Focus is on power, control and flexibility**
 - **No black boxes**
 - **Productive, but never at the expense of power and expressiveness**
- **ATL Server for native XML Web services**
- **Generates managed and unmanaged code**

Managed Extensions To Visual C++

- **It's still Visual C++**
 - Minimal extensions in the ANSI spirit
 - Nothing taken away
 - Does everything Visual C# does
- **Moving your existing code to the .NET Framework works**
- **Power and control**
 - Mix managed and native code
 - Mix managed and unmanaged data
 - Allows you to gradually migrate
- **Full access to the .NET Framework**

Summary

- **Select the language that is best for you**
 - **Visual Basic or VBScript developer → Visual Basic .NET**
 - **Visual C++ or Java developer → Visual C#**
 - **JScript developer → JScript or Visual C#**

Resources

- ***Basic Web Controls*** - Basic syntax used to create Web controls
<http://msdn.microsoft.com/library/dotnet/cpguide/cpconaspsyntaxforwebcontrols.htm>
- ***Language Innovations in .NET*** - A page with a little history, a little object-oriented programming, a little inheritance and even a little overloading
<http://msdn.microsoft.com/vstudio/nextgen/technology/language.asp>
- ***A Public Newsgroup for some Language Answers*** - A solid newsgroup with many answers if you have the time to look
<http://msdn.microsoft.com/library/dotnet.languages.vb>
- ***Java User Migration Path to Microsoft .NET*** (JUMP to .NET)

Language Resources

- APL - <http://www.dyadic.com/>
- Cobol - <http://www.adtools.com/info/whitepaper/net.html>
- Component Pascal - <http://www2.fit.qut.edu.au/CompSci/PLAS//ComponentPascal/>
- Eiffel - <http://dotnet.eiffel.com/>
- Haskell - <http://haskell.cs.yale.edu/ghc/>
- Mercury - <http://www.cs.mu.oz.au/research/mercury/>
- Oberon - <http://www.oberon.ethz.ch/lightning/>
- Mondrian - <http://www.mondrian-script.org>
- Python - <http://aspn.activestate.com/ASPN/NET/index>
- Scheme - <http://rover.cs.nwu.edu/~scheme/>
- SmallTalk - <http://www.qks.com/>
- Standard ML - <http://research.microsoft.com/Projects/SML.NET/index.htm>

In-Depth Training On .NET Languages

Direct from Microsoft

- **2373A: Programming with Microsoft Visual Basic .NET (Prerelease)**
 - **Available August 2001**
- **2124A: Introduction to C# Programming for the Microsoft .NET Platform (Prerelease)**
 - **Available now on Visual Studio .NET Beta 1**

Microsoft®